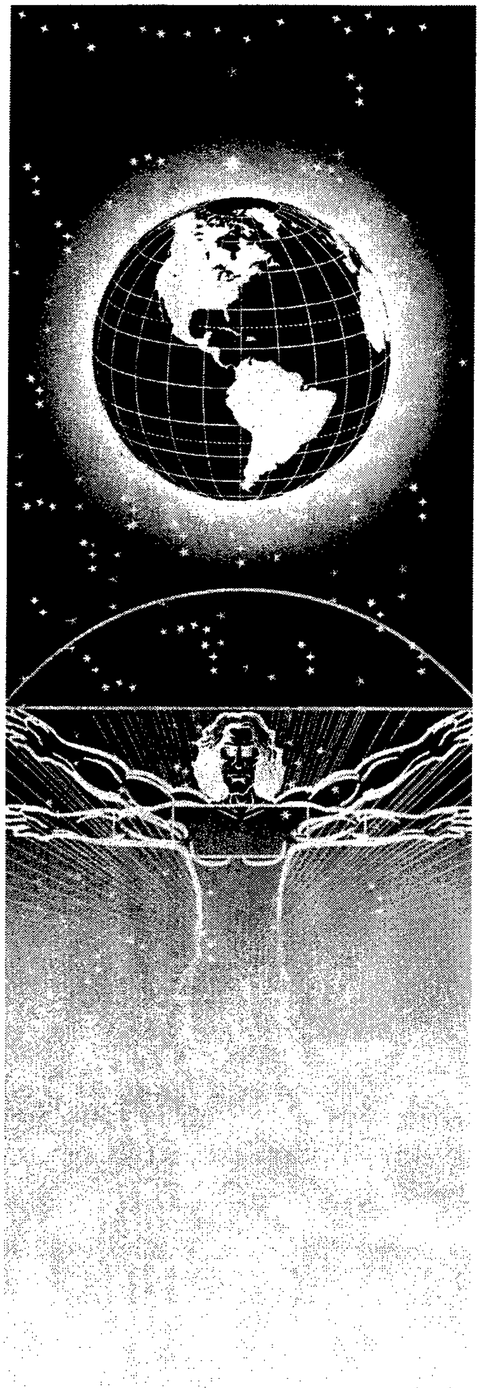
STINFO COPY

AFRL-HE-WP-TR-2003-0150



UNITED STATES AIR FORCE RESEARCH LABORATORY

IMPROVED BEHAVIORAL REPRESENTATION FOR OPERATIONS OTHER THAN WAR WITHIN AGGREGATE LEVEL SIMULATIONS

Nils D. LaVine
Steven D. Peters
Lee Napravnik

MICRO ANALYSIS & DESIGN, INC.
4949 PEARL EAST CIRCLE, SUITE 300
BOULDER CO 80301

NOVEMBER 2003

FINAL REPORT FOR THE PERIOD JUNE 2001 TO JULY 2002

Approved for public release; distribution is unlimited

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB, OH 45433-7022

NOTICES

DISCLAIMER

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2003-0150

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

//Signed//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 2003	Final, June 2001 – July 2002

4. TITLE AND SUBTITLE

Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations

5. FUNDING NUMBERS

C: F41624-98-C-6012
PE: 63231F
PR: 2830
TA: 29
WU: 40

6. AUTHOR(S)

Nils D. LaVine, Steven D. Peters, Lee Napravnik

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Micro Analysis & Design, Inc.
4949 Pearl East Circle, Suite 300
Boulder, CO 80301

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-HE-WP-TR-2003-0150

11. SUPPLEMENTARY NOTES

Air Force Project Manager: Mr. Gregory Barbato, AFRL/HECI

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This technical effort was sponsored by the Defense Modeling and Simulation Office (DMSO) and performed over a 14-month period under Army Research Laboratory (ARL) contract DAAD17-00-A-5003, "Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations." This program will be referred to as the Operations Other Than War (OOTW) Human Behavioral Representation (HBR) program.
This report summarizes the objectives and outcomes of this program. The focus of this effort was to implement improved OOTW HBR into a constructive simulation. For this effort we reviewed various constructive simulations to evaluate their ability to portray human behaviors in an OOTW setting, and based upon this review, a candidate constructive simulation was selected. We then cataloged currently available HBR behaviors within our selected OOTW constructive simulation. Next, within the constructive simulation, we implemented an advanced client-server architecture to incorporate improved HBR via an external server. We also developed and demonstrated a proof-of-concept OOTW HBR server using the client-server architecture.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Human Performance Model, Human Behavior Representation, Computer Generated Forces	33
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. Z39-18

This page left blank intentionally

TABLE OF CONTENTS

This page left blank intentionally

INTRODUCTION

PROJECT OVERVIEW

This technical effort was sponsored by the Defense Modeling and Simulation Office (DMSO) and performed over a 14-month period under Army Research Laboratory (ARL) contract DAAD17-00-A-5003, "Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations." This program will be referred to as the Operations Other Than War (OOTW) Human Behavioral Representation (HBR) program.

This report summarizes the objectives and outcomes of this program. The focus of this effort was to implement improved OOTW HBR into a constructive simulation. For this effort we reviewed various constructive simulations to evaluate their ability to portray human behaviors in an OOTW setting, and based upon this review, a candidate constructive simulation was selected. We then cataloged currently available HBR behaviors within our selected OOTW constructive simulation. Next, within the constructive simulation, we implemented an advanced client-server architecture to incorporate improved HBR via an external server. We also developed and demonstrated a proof-of-concept OOTW HBR server using the client-server architecture.

REPORT OVERVIEW

This report provides the following about the OOTW HBR program:

- background information that focused this effort,
- the selection of the constructive simulation to be used for this effort,
- description of our client-server architecture,
- discussion of the OOTW behaviors that we implemented within the constructive simulation,
- description of our HBR model of the implemented OOTW behaviors,
- and discussion of future work and recommendations for functional improvements to other OOTW behaviors, the selected constructive simulation, and the client-server architecture.

BACKGROUND

ISSUES IMPACTING THE OOTW HBR PROGRAM

The intent of this program is to improve the ability to investigate and study OOTW by providing more accurate representations of human behaviors in aggregate level simulations. Various issues influenced the direction that this program took to undertake this task. First, there are many different types of military operations that comprise OOTW. These generally fall into the categories of peacekeeping, peace making, and humanitarian relief. In this project, we felt it was imperative to select an aspect of OOTW that was relevant to today's real world operations, was realistically capable of being simulated, was of definite interest to the modeling and simulation community, and supported other DMSO and non-DMSO sponsored efforts. Military Operations in Urban Terrain (MOUT) was selected as the area of focus within a constructive simulation that

would fulfill these requirements and would provide an excellent opportunity to demonstrate this improved capability. The following paragraphs discuss the factors that led to the decision to focus on MOUT for improving OOTW HBR.

First we will look at the categories of OOTW and their relevance to the DoD. Throughout the world the US military is currently involved with peacekeeping, peace making, and humanitarian relief operations that require extensive involvement of soldiers, sailors, marines, and airmen. We believe that any of these areas would provide an immediate benefit to both the modeling and simulation community and the DoD.

Drawing on our experience with modeling human behavior in constructive simulations, we believe that each of the three categories can effectively be simulated to provide meaningful insight into OOTW operations. Humanitarian assistance lends itself to simulations centered about logistical type operations. In humanitarian assistance, many of the issues of interest involve moving supplies to the correct location in the right amount of time and therefore usually involve time based simulation analysis. Peace making and peacekeeping operations lend themselves to more of the traditional combat simulations, particularly those involving small unit operations. .

During the research phase of this effort, we came into contact with many DMSO as well as non-DMSO sponsored programs that had an interest in OOTW operations. Of particular interest to many of these programs were MOUT operations. In particular, the DMSO programs associated with the Smart Sensor Web (SSW) were centered on MOUT operations. We found that not only was DMSO involved with studying the effectiveness of SSW technologies, but also was sponsoring a sister program to collection data on the capabilities of both SSW equipped soldiers and traditionally equipped soldiers in a MOUT environment. By focusing this program on OOTW MOUT operations, we could leverage the data collection from live exercises to improve constructive simulations through the OOTW HBR capabilities that we developed in this program. As an added benefit, we believe that the dual use of this data can greatly enhance the utility of constructive simulations for training, advanced concept exploration, requirements generation, and Tactics, Techniques, and Procedures (TTP) development.

While all three categories of OOTW operations have interest in the modeling and simulation community, OOTW in MOUT was consistently mentioned as an area of interest. U.S. forces are deployed in many different areas of the world performing different missions and a large number of them involve MOUT operations. Also, with the advent of the attacks during 11 September 2001, focus on MOUT and anti-terrorist operations has become more relevant. We have selected OOTW MOUT operations for this effort in direct support of this increased focus.

REVIEW OF CGF SYSTEMS

This section describes our selection of a constructive simulation for this effort. In this project, we first evaluated constructive simulation systems. Because the domain of interest was in OOTW, and specifically MOUT, we focused our review on Computer Generated Force (CGF) software systems that could already simulate MOUT operations. This narrowed our focus from all CGF applications that were directed at the highly aggregated level, such as campaign level simulations including WARSIM, to CGF applications that could represent operations at the Dismounted Infantry (DI) level of

fidelity. The following are the CGF systems that we identified as candidate CGF systems for this project:

- ModSAF (Modular Semi-Automated Forces (SAF)) version 5.0
- OneSAF Testbed Baseline (OTB) version 1.0
- JointSAF (JSAF) version 5.7B
- Dismounted Infantry SAF (DISAF) version 7.1
- Integrated Unit Simulation System (IUSS)
- Joint Combat and Tactical Simulation (JCATS)

For us to perform an in depth analysis of the MOUT capabilities contained in each of the above identified CGF systems required us to obtain the source code. With the exception of IUSS and JCATS, we were able to obtain source code and review both the software architecture and the CGF system's capability to be model operations in a MOUT environment. Without the source code for IUSS and JCATS we were unable to evaluate their MOUT capabilities and thus they were not considered as the constructive simulation for this effort.

After careful review, we selected DISAF as our constructive simulation for incorporating improved OOTW HBR. We developed a position paper of this CGF review early in this project and this position paper is provided in Appendix A. It concludes that DISAF is best suited to meet the needs of this program due to it's modeling of MOUT operations and dismounted infantry personnel, and it's availability and acceptance in the user community. That is, it is a constructive simulation that contains models of MOUT operations and is well grounded in the CGF modeling and simulation community.

BEHAVIORS WITHIN DISAF

As part of this project, we reviewed Dismounted Infantry (DI) behaviors within DISAF that would provide a good demonstration of improved behavioral representation. The results of this review can be found in Appendix B, "DISAF Behaviors for OOTW HBR Program".

The initial review was a broad survey of DI behaviors within the SAF, with a specific focus within each library on the existing task parameters, and the potential for modifying and improving their representation via an external server. While a large number of behaviors were examined, a large percentage of them were in an "open-field" category, which generally had no direct relation to MOUT activities. Given the MOUT focus taken for this project, these behaviors were eliminated for further consideration and examination of the existing capabilities proceeded. What was discovered was that the MOUT behaviors were very scripted, and there was little or no actual decision activity. While this level of modeling does provide a basic capability it is extremely lacking in terms of realism and human behaviors that can significantly influence the operation. Starting from this basic behavior framework, a decision was made to expand upon an existing behavior and provide it a greater degree of realism by implementing "hooks" into the task behavior so that a server could provide a dramatically increased level of intelligent control over the SAF entities.

Of the many behaviors with at least rudimentary implementation within DISAF, the behavior involved with DI entities clearing rooms provided the best groundwork for meeting this program's objectives. Specifically, we wanted to enhance the behaviors associated with DI entities performing a task of clearing a room. This task (or behavior) involves the DI soldiers stacking outside of the room, performing actions that would allow them to gain Situational Awareness (SA) of the contents of the room, and then entering the room using Rules of Engagement (ROE) based upon the SA that they had gained. The original SAF behavior had the user specify a single room within a building that the DI was to clear, and one of the sides of the room to stack on once they were inside. Once the order to perform the task was given, the unit would rush directly to the building entrance from wherever they were at the time, (5 feet or 5 miles away, it made no difference), and enter the room with "tight" fire permissions. (Tight would indicate to not engage an enemy until they had first been engaged.) They would then all proceed to a stacking location in the room, change fire permission to free (fire as soon as possible), and run with guns blazing to each DI's pre-programmed final position.

Since this lacked a great deal of realism, we decided to enhance the behavior. To this end, we added several pieces of functionality, most notably the concept of a delay while gathering intelligence prior to entering the room, and then entering (or not) the building in a manner appropriate to the gathered intelligence. First, a stack point outside the building was added so the unit would have a place to gather somewhere outside the building prior to entering. This provided two things, 1) the unit was now guaranteed to be together as a unit when entering the building, and 2) this provided an opportunity to simulate the gathering of intelligence prior to rushing in. Once at the outer position, the unit pauses for an amount of time that represents an intelligence gathering effort using intelligence equipment that could be specified by the model builder (infrared, looking through a window, camera, etc…). Based upon the provided intelligence the unit then decides the appropriate course of action to take to clear the room. This provides a much more realistic model of a DI unit performing a clear room operation in a MOUT environment.

To accomplish the complex and variable decision making process that is involved with this clear room operation, an external server was used. This server is connected to the client DISAF to provide improved HBR models to operate in conjunction with the constructive simulation. Using this architecture, the clear room operation can be modeled without affecting SAF performance and yet provide a realistic and useful representation of the tasks involved with clearing a room.

During the initial stages of clearing the room, the unit gathers at a stack point outside the building and pauses for a moment. The amount of time that they wait is simulated in the external server and that time is provided to the SAF. During this delay, the SAF has sent the server exact characteristics of all entities within the building with regards to force alignment. Using this information the number of enemy, friendly, and neutral entities in the building, as well as the number of living members in the assault team, the model can determine the action to take upon entering. In addition, an error modifier to the true intelligence can be modeled to mis-classify certain entities, and thus alter the actions the unit takes upon entering. This provides a more realistic representation of clear room operations. Thus, based upon the provided intelligence, and generated errors, and the

numbers of assault DI team members available, the model returns an action to take to the SAF. Sample actions are to abort the mission, toss in a fragmentation grenade and then charge, or charge in with hold, tight, or free fire permissions to immediately engage the enemy. With these modifications, much more realistic behaviors, as well as potential for catastrophic errors (very real), occur.

DISCUSSION

IMPLEMENTATION OF IMPROVED OOTW HBR

Client-Server Architecture

The approach that we used to improve entity behaviors within DISAF was to use a client-server architecture. The client-server architecture can facilitate the inclusion of both variable fidelity entity behavior and much more complex entity behaviors than is currently possible within DISAF. This client-server architecture allows entity behavioral representation to be off loaded from DISAF (the client) to an external behavioral representation server – the Behavior Server. Figure 1 shows an example federation that utilizes the client-server architecture. This approach has been implemented in multiple programs and documented in papers and reports presented to the simulation and modeling community [1,2,3].

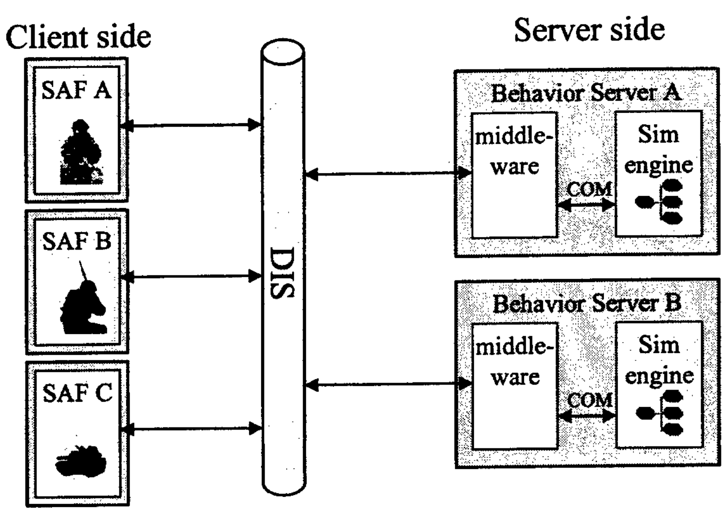


Figure 1. Client-Server federation.

The reasons for implementing a client-server architecture are based upon limitations exposed in our review of capabilities of current CGF systems [4]. Traditionally, the approach to improving behavioral representation within CGF systems has involved embedding additional software directly into the CGF software code base. Most likely, the improved behavioral representation algorithm is more complex and more computationally intensive. By embedding these complex software algorithms into the CGF system the software has been made more complex and the processing burden within the CGF system has been increased. This adversely affects its performance and intended purpose of simulating large numbers of entities.

A concept that alleviates these issues is to provide external processing capabilities by implementing a client-server architecture. To incorporate a client-server architecture, three software modifications are required of the DISAF. They are:

- The inclusion of software libraries that allow a subscription process to occur.
- Data handling libraries that send data requests to the server and then funnel the data responses back to the correct location within DISAF for other libraries to access.
- Modifications to situational response libraries to utilize the server-provided data.

Figure 2 shows both the client (DISAF) software modifications and the server (HBR) software architecture. We will now briefly discuss each of these client software modifications.

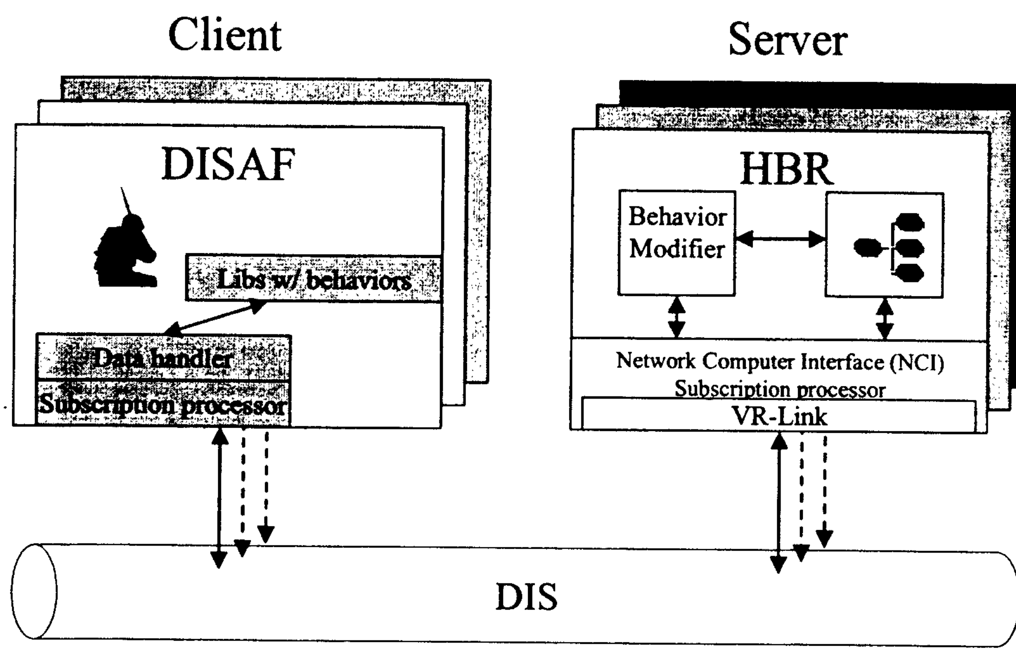


Figure 2. Client-Server architecture.

A software library that implements a subscription process will connect a DISAF entity to a server for a particular behavioral task. The subscription of an entity to a behavior provides a one-to-one mapping between a client SAF controlling the entity and a HBR server that provides that behavior to that entity. Using a Graphical User Interface (GUI), a DISAF user selects the desired entity or entities that should attempt to use a server. When the exercise (or scenario) is started, a subscription process occurs in which subscription requests to servers are sent out via Distributed Interactive Simulation (DIS) Protocol Data Units (PDU). Behavioral servers capable of providing that type of subscription request would then respond with appropriate action response DIS PDUs. This subscription process also includes load balancing between servers, and allows for multiple servers. The subscription hand-shake process between DISAF (the client) and

an OOTW HBR (the server) involving a pair a Action Request and Action Response interactions is shown in Figure 3.

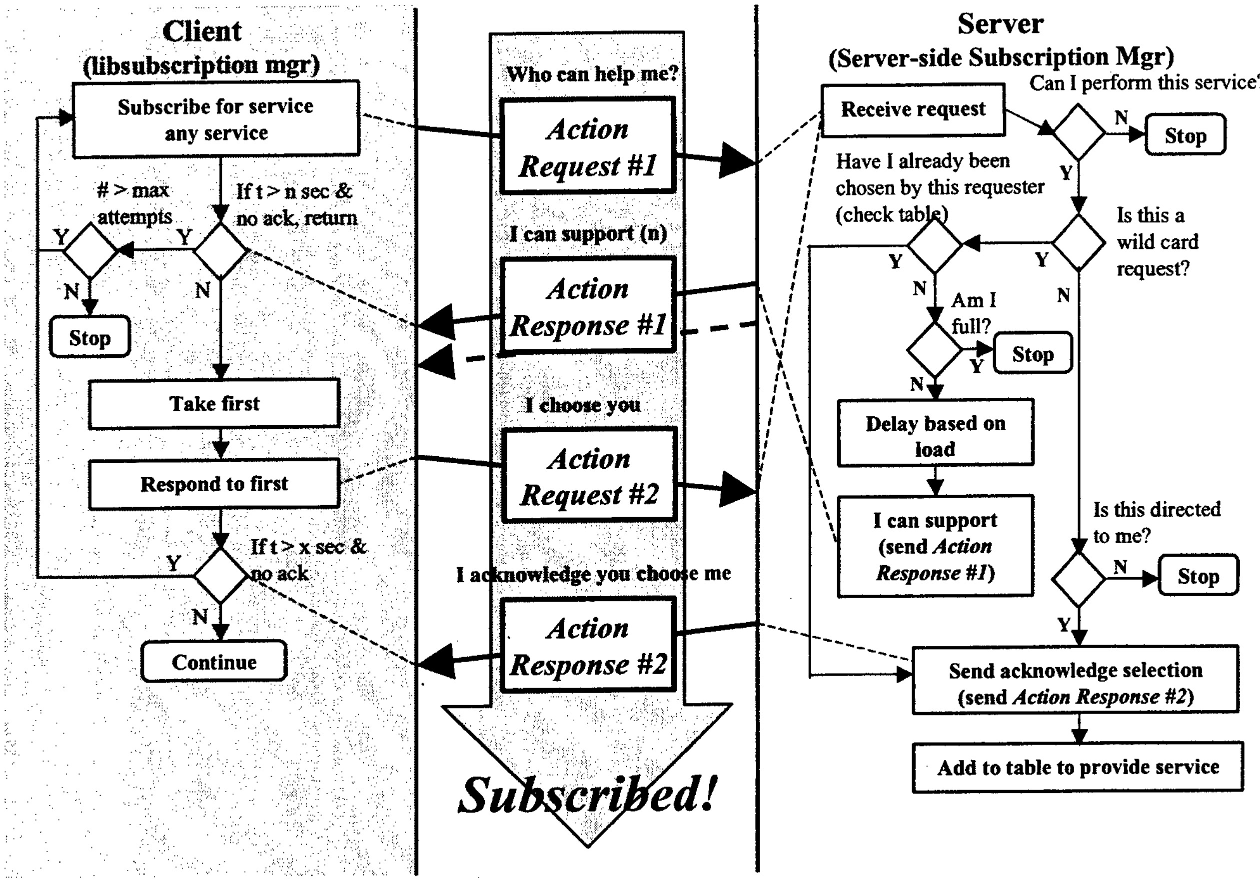


Figure 3. Client-Server subscription process.

The second type of software modification required of the DISAF involves the data transferred between the client and the server. A data handling software library is needed for sending requests for data to the server, receiving the data responses, and then putting the server provided information in a location accessible to the appropriate software behavior libraries. This request/response network communication between the client and server is conducted via DIS Data PDUs.

Lastly, the desired behavioral representation libraries within DISAF are modified to use the server provided information. This requires that behavior libraries be aware of and utilize server provided information. These libraries also trigger the data handler libraries to send out requests for information during a scenario exercise. Figure 4 shows this data handling methodology. For this effort we modified the unit clear room DISAF library (libuicclearroom) for improved MOUT capabilities using OOTW HBR.

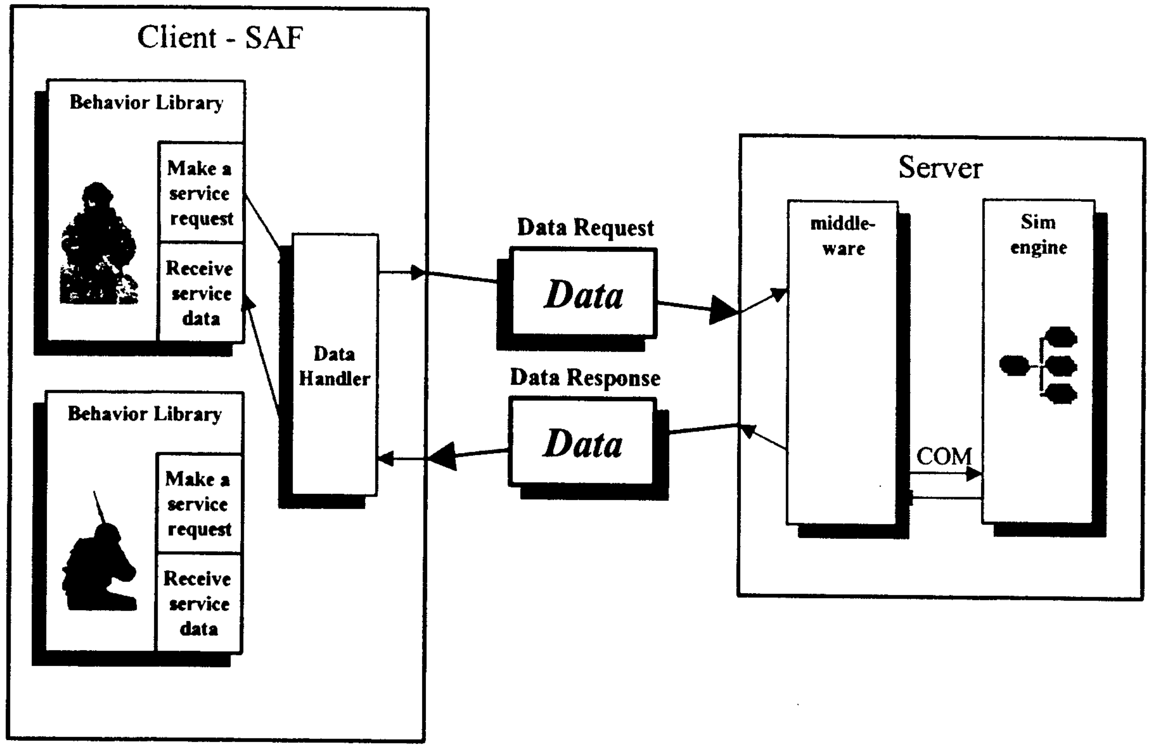


Figure 4. Client-Server data handling methodology.

MA&D has incorporated the use of the client-server approach into DISAF. Using this approach, we were able to include much higher fidelity behaviors into the CGF system without degrading its system performance. If we had attempted to embed this functionality directly into DISAF, it would have resulted in a significant negative impact on performance, possibly not meeting the requirement of maintaining realtime computational operations. Also, through this effort, we have laid the basic infrastructure for continuing to improve many other OOTW HBRs within DISAF. The subscription and data handling libraries are generic and can continually be used for new behaviors. The only modification required is within the existing behavioral libraries already within DISAF. Thus we can leverage this existing work to easily continually to improve many other DISAF entity behaviors.

OOTW HBR Models

With DISAF modified to take advantage of the client-server architecture and server software developed that can allow DISAF entity subscription and data handling, the final piece for this effort was to develop some basic OOTW HBR models. Our focus for this portion of the program was not to develop "definitive" OOTW HBR models, but to select a representative example to build a software infrastructure within a constructive simulation to allow continued improvement of OOTW HBR. For that reason, the OOTW HBR models we developed are somewhat basic and un-validated, yet provide a significantly better OOTW HBR than currently exists. The more significant point being that this architecture would provide a proof-of-concept of an ability to improve OOTW HBR within a constructive simulation through the inclusion of the client-server software infrastructure and external models easily modifiable into high fidelity HBRs. We believe

that future work in this area should focus on developing more realistic OOTW HBR models for the server and expanding upon the behaviors that we can influence.

For this OOTW HBR model development effort we choose a representative behavior involved with MOUT operations. Our modifications of DISAF's libuicclearroom were to incorporate intelligence gathering behaviors prior to DI entities entering a room and making decisions about room entry procedures and entity ROE based upon intelligence gathered

We developed two OOTW HBR models. When completed, these models reside within the HBR server and receive information from and send information to DISAF. The two separate models that we developed affect the behaviors within libuicclearroom. Each of the two models is broken down into a time delay to gather intelligence outside the room the DI's will enter and the a decision on how to enter the room based upon the intelligence that the DI's gathered outside the room.

The two models were developed within the Micro Saint task network modeling simulation package. One is a baseline case and in the other is an advanced technology case. In the baseline case, the DI entities have only current rudimentary capabilities. They have no advanced technologies that will allow them to actively or passively collect information about what is inside the room they are about to clear. Figure 5 shows the task network diagram of the DI entity actions as they "stack" outside of the room they will enter. Each task happens in sequential order and will take a amount of time. Thus, adding these task times together will determine that total task time required to perform the "Evaluate Room" task. Table 1 shows the task times associated with the DI entities performing the tasks.

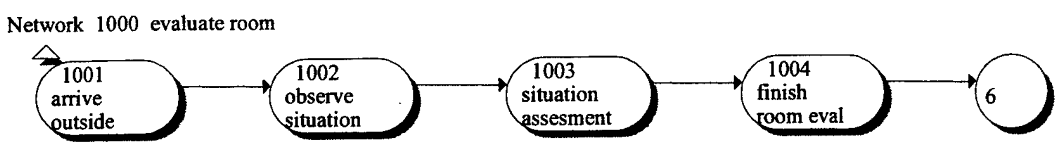


Figure 5. Baseline "Evaluate Room" task network diagram.

Table 1. Task Time means & standard deviations for baseline "Evaluate Room" task network diagram.

Task Name	Mean time (seconds)	Standard Deviation
Arrive Outside	0	0
Observe Situation	5	2
Situation Assessment	4	1
Finish Room Eval	0	0

For determining the room entry actions based upon the intelligence they gathered in the "Evaluate Room" task network diagram, DI entities make requests to the "method to clear room" task network diagram. Figure 6 shows the "Method to Clear Room" task network diagram. All tasks in this network diagram take no time and simply model the decision itself.

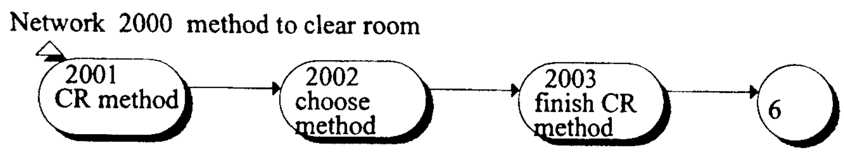


Figure 6. Baseline "Method to Clear Room" task network diagram.

In this portion of the model, the determination is made on room entry actions and ROE. For the baseline case, since there is no intelligence gathered other than waiting outside the room, we probabilistically determine how the entities will behave. Basically, these probabilities are broken down into a 1/3 chance of entering the room with a ROE of "hold", a 1/3 chance of entering the room with a ROE of "tight" and a 1/3 chance of entering the room with a ROE of "free". The following is the model logic for determining this:

```
urand := random();
if (urand <= 1/3) then
  entryAction[tag] := CHRG_IN_HOLD
else if (urand <= 2/3) then
  entryAction[tag] := CHRG_IN_TITE
else   entryAction[tag] := CHRG_IN_FREE;
```

Where random() is a function that returns a uniform random number between 0 and 1.

In the advanced technology case, the DI entities will have room sensing equipment as is being investigated in the DMSO Smart Sensor Web (SSW) program. In our SSW HBR "Evaluate Room" model, we provide the capability for using either a daylight TV or a Forward Looking Infrared Radar (FLIR) to gain intelligence about the room. Figure 7 shows the task network diagram of the DI entity actions as they "stack" outside of the room. Table 2 shows the task times associated with the DI entities performing the tasks.

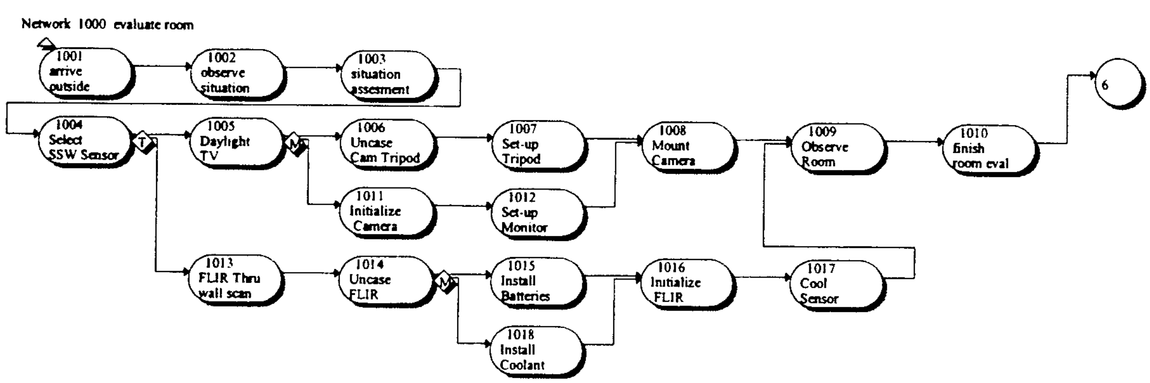


Figure 7. SSW "Evaluate Room" task network diagram.

Table 2. Task Time means & standard deviations for advanced technology "Evaluate Room" task network diagram.

Task Name	Mean time (seconds)	Standard Deviation
Arrive Outside	0	0
Observe Situation	5	2
Situation Assessment	4	1
Select SSW Sensor	2	.5
Daylight TV	0	0
Uncase Cam Tripod	5.5	1.1
Set-up Tripod	12.5	2.5
Mount Camera	4.5	1.1
Observe Room	10	1.66
Initialize Camera	3.5	1
Set-up Monitor	15	1.5
FLIR Thru wall scan	0	0
Uncase FLIR	6.2	1
Install Batteries	2.5	.75
Initialize FLIR	3.5	.75
Cool Sensor	15	0
Install Coolant	3	.5
Finish Room Eval	0	0

For determining the room entry actions based upon the intelligence gathered in the "Evaluate Room" task, DI entities make requests to the "method to clear room" task network diagram just as in the baseline case. Figure 8 shows the "Method to Clear Room" task network diagram. All tasks in this network diagram take no time.

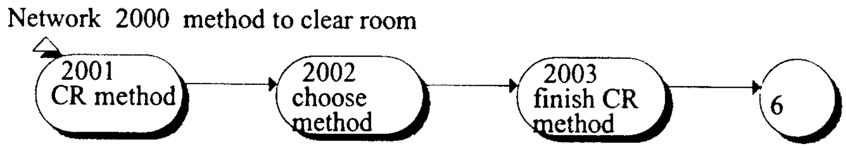


Figure 8. SSW "Method to Clear Room" task network diagram.

In this portion of the model, the determination is made on room entry actions and ROE. For the baseline case, since there is no intelligence gathered other than waiting outside the room, we probabilistically determined how the entities would behave. In this case, since we can gain intelligence through advanced technologies, we also "fuzzy" the gathered intelligence to model imperfect intelligence gathering capabilities. For the FLIR, we assigned a 95% probability of gathering perfect intelligence. For the other 5% of the time, there is a chance that the intelligence is skewed by up to 30%. The following is the model logic for determining this:

```
urand := random();
if (urand <= .95) then
  sensEnemy[tag] := numEnemy[tag],
  sensFriend[tag] := numFriend[tag],
  sensNeutral[tag] := numNeutral[tag];
else
  if (random() <= 0.5) then
    sensEnemy[tag] := round(sensEnemy[tag] * -0.30 * random()
  else
    sensEnemy[tag] := round(sensEnemy[tag] * 0.30 * random()
  if (random() <= 0.5) then
    sensFriend[tag] := round(sensFriend[tag] * -0.30 * random()
  else
    sensFriend[tag] := round(sensFriend[tag] * 0.30 * random()
  if (random() <= 0.5) then
    sensNeutral[tag] := round(sensNeutral[tag] * -0.30 * random()
  else
    sensNeutral[tag] := round(sensNeutral[tag] * 0.30 * random()
```

Based upon this "fuzzied" intelligence, the OOTW HBR will determine the room entry procedures. As opposed to the baseline case, since we now have intelligence to base this decision on, this now becomes much more complicated. In this case, we determine entry actions as based upon friendly, enemy, and neutral personnel in the room and also the number alive our room entry unit. If we are outnumbered by enemy elements in our unit, we choose to abort the room entry operation. If we have more members in our room entry unit and the number of enemy in the room exceeds the number of friendly and neutral elements in the room, then we enter the room with a ROE of "free". If we have more members in our room entry unit and the number of enemy in the room is less than the number of friendly and neutral elements in the room, then we enter the room with a ROE of "hold". The following is the model logic for determining the room entry actions:

```
        if sensEnemy[tag] >= numAlive[tag] then
entryAction[tag] := ABORT
```

```
else if ((sensEnemy[tag] < numAlive[tag]) & (sensEnemy[tag] > (sensFriend[tag]
+ sensNeutral[tag]))) then
  entryAction[tag] := CHRG_IN_FREE
else if ((sensEnemy[tag] < numAlive[tag]) & (sensEnemy[tag] <  (sensFriend[tag]
+ sensNeutral[tag]))) then
  entryAction[tag] := CHRG_IN_HOLD;
```

DISAF makes requests for this information and send SA information such as the number of friendly, enemy, and neutral elements in the room along with the number of living members in the room entry unit. In response to those requests, these OOTW HBR models calculate and send entry action commands and room evaluation times to DISAF.

CONCLUSION

In this effort for DMSO, MOUT operations were chosen for study due to their importance in current world affairs, their inadequate level of modeling in constructive simulation, and their ability to gain benefit from our client-server architecture. DISAF was selected as the constructive simulation to develop and demonstrate MOUT behaviors. A behavior in DISAF in which a unit of soldiers clear a room was connected to an external server that could execute one of two task network models of that behavior. One model provided a baseline representation in which soldiers gather outside a room and then enter it, shooting at every enemy in view. The other model provided a more advanced technology approach to clearing a room by using devices outside the room to improve the situational assessment of the room and affect the rules of engagement for entering the room.

DEMONSTRATION

We have implemented the client-server architecture within both client and server applications. We have also developed a DISAF room entry scenario along with two Micro Saint OOTW HBRs (the time to spend outside the room while gathering situational assessment knowledge, and the rules of engagement to use upon entering the room) for the MOUT environment. We demonstrated this capability at the 10 July 2002 DMSO S&T/BAA review conducted at DMSO Headquarters in Alexandria, VA. We provided a presentation of our capabilities that includes many of the graphics in this final report. We also provided a real time demonstration of our client-server architecture operating with DISAF and an OOTW HBR server. DMSO personnel agreed that this demonstration provided needed capabilities for constructive simulations, HBR, and MOUT in OOTW.

FUTURE EFFORTS

In this instance, we implemented room-clearing behaviors to include "stacking" and gathering intelligence prior to room entry and also the method and ROE for room entry. While we believe that we have successfully demonstrated the capability of improving OOTW HBR within a constructive simulation, we believe that there is a significant amount of future work that we can still be accomplish. One area of future development would involve implementation of more behaviors within the SAF.

Another area is in the development of OOTW HBR within the server. DMSO conducted a data collection exercise with live participants at the McKenna MOUT site at Ft.

Benning, GA in March of 2002. We believe that this as well as other exercises is an excellent opportunity to utilize live simulation data to improve constructive simulations.

Another area of future work is improving constructive simulation software architecture. In MA&D's report titled "An Advanced Software Architecture for Behavioral Representation within Computer Generated Forces" we address issues associated with constructive simulation architectures that limit their effectiveness for training, advanced concepts exploration, and weapon system analysis [4]. These issues include: behaviors within CGF systems are difficult to discover; entity behaviors can be affected by multiple software libraries; CGF system code bases are continually growing in side and becoming more complex; there is a limited capability for modifying entity HBR; CGF systems have poor temporal resolution; entity foundations are not based upon their particular entity type, and the onus for Validation, Verification, and Accreditation (VV&S) is on the CGF sponsoring organization. These issues need to be addressed and improved upon in order to gain the full potential of constructive simulations.

BIBLIOGRAPHY

[1] LaVine, N., Kehlet, R., O'Connor, M., and Jones, D., "Transferring Ownership of ModSAF Attributes", Proceedings of the 1999 Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL Spring 2002.

[2] LaVine, N., Kehlet, R., and Peters, S., "A Client-Server Approach to CGF Behavioral Representation", Proceedings of the Eighth Conference on Computer Generated Forces & Behavioral Representation, Orlando, FL, May 1999.

[3] S.D. Peters, LaVine, N.D. and Napravnik, L.J., "Composable Behaviors in an Entity Based Simulation", Proceedings of the 2002 Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL March 2002.

[4] N.D. LaVine, and S.D. Peters: "An Advanced Software Architecture for Behavioral Representation within Computer Generated Forces", Final Report, Defense Modeling and Simulation Office, Alexandria, Virginia, December 2001.

APPENDIX A

POSITION PAPER: AGGREGATE LEVEL SIMULATION SELECTION

Purpose: The purpose of this paper is to provide the selection of an Aggregate Level Simulation (ALS) for the Defense Modeling and Simulation Office (DMSO) Science and Training Initiative Directive (STID) for fiscal year 2001 Simulation and Training (S&T) call titled "Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations." This project will be referred to as Operations Other Than War (OOTW) Human Behavior Representation (HBR) for the rest of this paper.

Background: A major problem indicated by users of simulations, especially distributed simulations, has been the lack of realism of simulated entities and units. Another problem is the lack of available behaviors for a user to perceive or interact with. If constructive simulations are to be used to properly train warfighters; determine correct tactics, techniques, or procedures; test and evaluate a potential weapon system; or perform different types of mission rehearsal, then improved human behavior representation within constructive simulation is a necessity.

The OOTW BR program will address the issue of behavior representation by developing a client-server architecture between an ALS and the Combat Automation Requirements Testbed (CART) Human Performance Modeling Environment (HPME). This modification of the architecture will allow high-fidelity human performance models to provide entity behavioral representation parameters to the ALS during runtime. This client-server relationship will provide the benefit of being able to change behavioral representation for selected entity behaviors without either having to change CGF system software or degrading the CGF system's runtime performance.

Discussion: In this project, we first evaluated ALS systems. Because the domain of interest was in OOTW, and specifically Military Operations in Urban Terrain (MOUT), we focused our review on Computer Generated Force (CGF) software systems that could simulate MOUT operations. This narrowed our focus from all CGF applications that were directed at the highly aggregated level, such as campaign level simulations including WARSIM, to CGF applications that could represent operations at the Dismounted Infantry (DI) level of fidelity. The following are the CGF systems that we identified as candidate CGF systems for this project:

- ModSAF (Modular Semi-Automated Forces (SAF)) version 5.0
- OneSAF Testbed Baseline (OTB) version 1.0
- JointSAF (JSAF) version 5.7B
- Dismounted Infantry SAF (DISAF) version 7.1
- Integrated Unit Simulation System (IUSS)
- Joint Combat and Tactical Simulation (JCATS)

Based upon the identification of the above CGF systems, we attempted to obtain source code for further evaluation. In all but two instances, we were able to obtain source code and review both its software architecture and the CGF system's capability be used in a

MOUT environment. We were unable to obtain source code for IUSS and JCATS. The following are reviews of each of the candidate CGF systems.

ModSAF

ModSAF is a CGF system that is the baseline from which many of the other CGF systems were developed. It has been and continues to be the CGF system employed by the US Army. Simulation, Training, & Instrumentation Command (STRICOM) is the U.S. government sponsoring organization for ModSAF. The ModSAF software libraries are written in the C programming language and run under the LINUX, IRIX, and Windows NT operating systems. ModSAF implements network communication using the Distributed Interactive Simulation (DIS) protocol and it can operate within an High Level Architecture (HLA) environment using a DIS-HLA gateway.

Many other CGF systems are derivatives of ModSAF and have inherited its behavioral representation system architecture. While these other ModSAF based CGF systems were developed for different purposes, their underlying behavioral representation architecture is identical.

Our evaluation of these various CGF systems has discovered some issues with the software architecture that adversely affect representation of behavioral effects. An architectural design decision in the development of ModSAF was to implement behavioral representation within modular software libraries. This architecture structure focuses on implementing in a modular fashion the algorithms associated with reacting to a stimulus and providing a proper situational response. These situational response libraries in turn affect entity parameters that are then acted upon by the entity. This architecture permits multiple software libraries to affect the same entity parameter. (By having a software data structure in which entity parameters are defined globally, any software library has access to entity parameters and can change them.) The architecture does not include a method for adjudicating situations where multiple software libraries provide varying inputs for the same entity parameter.

Traditionally, additional functionality and modification to ModSAF has been made directly into the software code base. This method of development has resulted in ModSAF version 5.0 containing over 1.1 million lines of code in 582 software libraries. This ever-increasing code base has become complex and cumbersome and requires significant expertise in both programming and ModSAF in order to modify or add to it.

Of the CGF systems evaluated for this program ModSAF has the most limited capability to portray DI entities. Since all of the other CGF systems have ModSAF's capabilities, this was ruled out as a candidate for our ALS.

OTB

OTB is the successor to ModSAF for the Army's mainstream CGF system. STRICOM is also the sponsoring organization OTB. OTB will bridge the gap of CGF systems between ModSAF and the OneSAF Objective System (OSS). The purpose of OTB is to evaluate new technologies and functionality that the user community desires from its CGF systems. If an idea has merit, it would likely become a candidate for incorporation into the OSS.

Because OTB is a test bed, it is a tool still under development. Version 1.0 is an improved version of ModSAF 5.0 and as such remains DIS compliant. To run HLA a Gateway is required. OTB has similar programming language and operating system characteristics as ModSAF. OTB has about 1.7 million lines of code in 596 software libraries.

OTB grew directly out of the ModSAF Version 5.0 software and thus its underlying architecture is identical to ModSAF's. Behavioral representation and how it is implemented within the software architecture is the same as ModSAF's and thus contains the same issues as stated above. Currently, OTB's DI entity capabilities are slightly expand upon that ModSAF V5.0's. Each of the Individual Combatant (IC) entities listed below:

- US IC w/ SAW & Hand Grenade
- US IC w/ M16A2 & Hand Grenade
- US IC w/ AT8 & Hand Grenade
- USSR IC AK47

have the following mission level behaviors:

- halt
- mount ground/air unit
- dismount ground/air unit
- hasty occupy position, road march
- suppressive fire
- move
- pursue
- assault
- withdraw
- attack by fire.

JSAF

JSAF version 5.7B has been developed by the US Army and Navy and is a variant of ModSAF 5.0. The sponsoring organization is Joint Forces Command (JFCOM). JSAF was developed to provide entity representation in virtual environments for non-Army services. JSAF includes substantially more types of entities than ModSAF and includes associated behaviors for naval and air platforms.

Because JSAF is a variant of ModSAF, it too uses the modular situational response library approach to behavioral representation. JSAF does however have a native HLA capability. JSAF has similar programming language and operating system characteristics as ModSAF. JSAF has approximately 2.3 million lines of code in 842 software libraries.

JSAF has greatly expanded upon the DI capabilities that are available either within OTB or ModSAF. This is probably due to the Marine Corps influence in JSAF's development. Following is a list of available DI entity types within JSAF.

- USMC Flare Gun
- USMC Rifle Co Cdr
- USMC Rifle Plt Ldr
- USMC Rifle Sq Ldr
- USAF DI-Flight Cdr
- USAF DI-Flight Sgt
- USAF DI-RTO
- DI w/LAW80
- Civilian Non-Combatant
- Rifle Co Cmdr
- Rifle Co D Cmdr
- Rifle Co 1st Sgt
- Rifle Plt Ldr
- Rifle A. Plt Ldr
- Rifle Squad Leader
- Rifle Grenadier
- 12.7mm Medium MG Gunner
- Rifleman (AK47)
- Rifleman (AK47+RPG7)
- Weapons Sq Ldr
- 7.62mm PKm MG Gunner
- Medium MG A

- Mortar Plt Leader
- Mortar Squad Leader
- 60mm Mortar Gunner w/mortar
- Mortar Gwd Observer
- Rifleman (SA7)
- TOW Gunner

Also listed are the mission level behaviors available to the DI entities.

- Subordinate Tasking
- Report To
- Accept Unit
- Unassign Unit
- Assemble
- Suppressive Fire
- Breach
- Minefield Traverse
- Hide
- Unhide
- Simple Move
- Move Repairman
- Transit Tunnel
- Fast Unhide
- Unhide/Move/Rehide
- IC Move
- IC Follow Vehicle
- IC Pursue
- IC Halt
- IC Occupy Position
- Assault
- IC Embark
- Fire and Movement
- Displacement
- IC Place Satchel
- IC SOC w/AOC
- IC Hasty Occupy Position

While there are many more entity types in JSAF than found in ModSAF or OTB, JSAF treats this larger group of entities in a similar fashion. As an example, while a civilian entity may look visually different, it has the same behaviors as combatant entities.

DISAF

DISAF was developed to increase the capability to portray dismounted infantry on the virtual battlefield in a realistic fashion. The primary focus of DISAF has been the development of tactical behaviors for individual through squad level operations. DISAF is maintained by STRICOM out of Orlando, FL and is a variant of OTB version 1.0.

As a result of DISAF being based upon the OTB architecture, it has the same ModSAF software architecture and associated issues. It uses the same situational response approach for behavioral representation software libraries. DISAF can be networked using the DIS protocol or the DIS-HLA gateway. DISAF runs on a SGI under IRIX 6.2 or on a PC under Linux or Windows NT. DISAF has about 1.9 million lines of code in 620 software libraries.

Because DISAF's focus is on simulating DI entities, its capabilities for OOTW operations are greatly expanded upon than any of the other CGF system that we reviewed. DISAF has built-in support for MOUT operations and can make use of terrain databases that support the Multiple Elevation Surface (MES) structures. A terrain database of the McKenna MOUT site comes with DISAF.

DISAF also includes a 2D Plan View Display (PVD) especially modified for supporting DI entity operations, MES, and enhanced DI icons. DISAF also includes the capability for having a wider variety of non-combatants than any of the other CGF systems. The behaviors associated with the various types of DI entities also are more varied. As would be reasonable, non-combatants do not have all of the mission level behaviors associated with an infantryman. In addition, the non-combatants do have the mission level behaviors required to simulate terrorist actions. The following is a list of IC entities that are available within DISAF:

- US IC w/ M16A2 & Hand Grenade
- US IC w/ AT8 & Hand Grenade
- US IC w/ SAW & Hand Grenade
- US IC w/ M203 & Hand Grenade
- US IC Fireteam A (M16, AT8, M203, SAW)
- US IC Fireteam B (M16, M16, M203, SAW)
- US IC Fireteam C (M16 x 3, SAW)
- US IC Auto Weapons Team (M16 x 2, SAW)
- US IC Squad (M16, Fireteam A, Fireteam B)
- US IC Rifle Squad (M16, Fireteam B x 2)
- US IC Auto Weapons Squad (M16, Auto Weapons Team x 3)
- US IC Platoon (M16 x 2, Rifle Squad x 3, Auto Weapons Squad)

- USSR IC AK47
- USSR IC Squad (AK47 x 6)
- Civilians (man in suit, man in jacket, woman in suit, woman in jacket)
- Furniture (is furniture an entity?)

DISAF IC entities have the capability to perform individual and unit/team behaviors. U.S. DI entity behaviors are as follows:

- halt
- fire & movement
- throw grenade
- occupy position
- fire at location
- react to ambush
- suppressive fire
- react to contact
- move on path
- break contact
- mount/dismount
- clear room
- move tactically
- climb up/down

For team/unit actions, these same entity types can perform a fireteam clear room and squad clear room behavior.

Civilian entities within DISAF can do the following:

- halt
- climb
- move on exact path
- move to a point
- provide suppressive fire

- rush movement with fire
- throw grenades

They can also have autonomous behaviors of reacting to fire and wandering around the virtual battlefield. DI REDFOR DI entities within DISAF have the following behavioral capabilities:

- look around
- face bogey
- engage threat
- seek cover
- observe
- engage from cover
- fall prone & freeze
- freeze.

IUSS

IUSS was developed for the U.S. Army Soldier System Command at Natick Labs in Natick, MA. Due to the proprietary nature of its development, obtaining IUSS source code for this review was not possible.

JCATS

JCATS was developed by Lawrence Livermore Labs and is sponsored by the Joint Warfighting Center in Ft Monroe, VA. Due to the proprietary nature of its development, obtaining JCATS source code for this review was not possible.

Decision on ALS selection: Based upon the review of available CGF systems, we think that the DISAF CGF system best fulfills the needs for OOTW. We are selecting DISAF for the following reasons:

1. DISAF is specifically being developed for DI and MOUT operations. The objective of our project is to provide improved entity representation for OOTW operations in a MOUT environment. DISAF provides the best capability for assisting us in this effort. The behaviors available for DI entities within DISAF are much more aligned with the type of human performance models that we would build than any of the other candidate CGF systems.
2. DISAF comes with detailed terrain of the McKenna MOUT site. We know that we are going to be receiving data collected on the McKenna MOUT from DMSO's Smart Sensor Web (SSW) data collection effort. Because we will have the McKenna MOUT site terrain database, when we build human performance models we will be able to validate the models based upon the SSW scenarios. Also, as human performance models of new capabilities are developed, having the

McKenna MOUT site terrain will allow analysis of concepts that may be of interest to the SSW program.

3. From DMSO's SSW program kick-off meeting at Ft. Benning, GA in June, we coordinated with NAWC-TSD, whose work in the DI CGF area DMSO is interested in, and they are also work with DISAF.
4. DISAF is a derivative of OTB and thus supports the OOS.

We believe that DISAF provides the best CGF system for the "Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations" program.

APPENDIX B

INFORMATION PAPER: DISAF BEHAVIORS FOR OOTW HBR PROGRAM

Purpose: The purpose of this paper is to provide details on the available DISAF behaviors for the Defense Modeling and Simulation Office (DMSO) Science and Training Initiative Directive (STID) for fiscal year 2001 Simulation and Training (S&T) call titled "Improved Behavioral Representation for Operations Other Than War Within Aggregate Level Simulations." This project will be referred to as Operations Other Than War (OOTW) Human Behavior Representation (HBR) for the rest of this paper.

Background: A major problem indicated by users of simulations, especially distributed simulations, is the lack of realism of simulated entities and units. Another problem is the lack of available behaviors for a user to perceive or interact with. If constructive simulations are to be used to properly train warfighters; determine correct tactics, techniques, or procedures; test and evaluate a potential weapon system; or perform different types of mission rehearsal, then improved human behavior representation within constructive simulation is a necessity.

The OOTW HBR program will address the issue of behavior representation by developing a client-server architecture between a constructive simulation and the Combat Automation Requirements Testbed (CART) Human Performance Modeling Environment (HPME). This modification of the architecture will allow high-fidelity human performance models to provide entity behavioral representation parameters to the ALS during runtime. This client-server relationship will provide the benefit of being able to change behavioral representation for selected entity behaviors without either having to change CGF system software or degrading the CGF system's runtime performance.

Discussion: In this project, we first evaluated ALS systems. As a result of this effort, we selected DISAF as a constructive simulation that would best represent the OOTW domain, and specifically Military Operations in Urban Terrain (MOUT). We focused our review of OOTW HBR within DISAF on behaviors that are used in MOUT operations. This review is presented below. We conducted this evaluation by reading the code documentation where possible and reviewing and reverse engineering DISAF source code. While we did spend a significant amount of time conducting this review, it was not exhaustive. The DISAF OOTW behaviors that we did come into contact with are documented here.

DISAF Behavior Evaluation:

<u>Behavior #1</u>

Library name: libvthrowgrenade

Task description: This is an individual level task that controls the timing, location, and method by which a fragmentation grenade will be thrown at a specified target.

Task parameters:

1) location to throw the grenade
2) time to hold the grenade prior to throwing

3) throwing posture

Potential task alterations:

1) modify the length of time the grenade is held prior to throwing, possibly allowing for a long hold resulting in a "too close" explosion
2) add a force modifier to the throw to influence the bounce pattern of the grenade at the target
3) add modifiers to the actual targeting of the grenade, allowing for over/under throwing
4) if multiple types of grenades are simulated, allow for the choice of grenade to be based on the current situation.

Behavior #2

Library name: libvicicfiremovement

Task description: This is an individual level task controlling the method by which an individual combatant (IC) advances upon a location while providing fire on an enemy location. This is an open-field behavior and was not examined in depth for this study since the desired focus was on MOUT activities.

Task parameters:

N/A

Potential task alterations:

N/A

Behavior #3

Library name: libuicreacttoambush

Task description: A unit level task controlling if and when a unit decides it is "under serious fire", and what actions they take based on the current situation. Possible actions include taking cover and returning fire, occupy a position, and tossing grenades at the enemy.

Task parameters:

1) time under fire before the IC Unit decides that it is threatened and needs to react to the enemy.
2) enemy location

Potential task alterations:

1) modify the time it takes the unit to feel threatened
2) add "type" or "volume" of fire to the determination of serious threat

3) alter the reaction taken based on type/volume of fire, spotted enemy, number of entities in unit, etc..

4) allow for more reactions, such as continue mission while providing covering fire, advance upon enemy position, and withdraw.

<u>**Behavior #4**</u>

Library name: libvicclearroom

Task description: This is an individual level task that controls the actions an IC takes to clear a room of all threats. Actions include entering a room, moving to a securing positions, and engaging any potential enemy.

Task parameters:

1) stacking point that the IC advances to upon entering the room
2) route by which the IC advances to the stacking point
3) speed the entity advances at

Potential task alterations:

1) fire permissions the IC has upon entering the room
2) movement patterns
3) choice of entry points and secure positions based upon room and threat data
4) target prioritization based on current situation
5) ability to lay down suppressive fire

<u>**Behavior #5**</u>

Library name: libuicclearroom

Task description: This is a fireteam level task by which a unit advances into and secures a single user specified room. Clear Room is a unit level behavior designed specifically for an Army four-person fireteam to clear a room in a building. The behavior ends when ICs have reached their final positions in the room to be cleared.

Task parameters:

1) the room for the fireteam to clear

Potential task alterations:

1) modify unit fire permissions for entering the building
2) determine the route used to enter the building, and the stacking points inside the room, based upon room data such as dimensions, doorways and windows, or the number of enemy/neutral/friendly entities within the room.
3) allow the unit to clear not just the user-specified room, but the entire building

4) provide for actions taken prior to entering the building, such as intelligence gathering, assuming formations, or using explosive and/or non-lethal devices.
5) Specify the type of weapons to use upon entering the room
6) Allow the unit to pursue, acquire, and extract a "target of interest"

Behavior #6

Library name: libsqicclearrooms

Task description: Squad Clear Rooms is a unit level behavior designed specifically for two fireteams plus one Squad leader to clear rooms in a building. The behavior ends when the fireteams and the leader have reached their final positions in the room to be cleared. Room clearing is performed by control of posture (stance), weapon state, rules of engagement and movement. The unit level behavior assigns unique roles to the leader and each fireteam entering the building. The leader is the last man through the door, where he will mark it to signal that the room has been cleared once the fireteams have reached final positions

Task parameters:

1) initial room to enter/clear
2) first fireteam to enter the building
3) index of rooms to be cleared
4) stacking positions within the room

Potential task alterations:

1) allow the squad to clear the entire building
2) provide intelligence prior to entering
3) allow the squad to position themselves outside the building in a tactical manner, instead of just rushing from their current location to the entry point
4) allow for targets of opportunity based on gathered intelligence, such as enemy type/numbers/composition, available weapons and support, and other tactical considerations.
5) choose the entering fireteam based on available weaponry, experience, and task at hand.
6) clear the building in an intelligent, deterministic manner, instead of just a random sequential ordering

Behavior #7

Library name: libuclearroom

Task description: Clear Room is a unit level behavior designed specifically for an Army four person fireteam to clear a room in a building. The behavior ends when ICs

have reached their final positions in the room to be cleared. The behavior depends on the correct call sign assignments to team members. Each entity is distinguished by weapon type and call sign, and will first follow the given route through the entry door to a final position.

Task parameters:

1) ready and final positions of each entity within the fireteam
2) route the entities take through the door into the room
3) the room to be cleared
4) speed at which the entities advance into the room
5) fire permissions
6) number of living team members required to continue executing the clear room behavior

Potential task alterations:

1) provide intelligence to the unit prior to entering the building
2) choose ready and final positions based on intelligence
3) choose positions based on weapon types and expected threats
4) provide for actions to take prior to entering, such as use of area-of-effect weapons
5) alter the speed, posture, or ROE for the entities